IN THE SPECIFICATION:

In the specification, Applicants amend as follows:

Page 1, lines 14-25, first paragraph of the Background:

An integrated development environment (IDE) is a set of programs run from a single user interface. For example, programming languages often include a text editor, compiler and debugger, which are all activated and function from a common menu. One example of an IDE is VisualAge for Java, which is a product available from International Business Machines Corporation. Using an IDE, a developer is able to generate new software and programs from a single interface. Further, ~~and~~ an IDE also may provide a graphical user interface for various programming features to increase the ease of creating software and programs.

Page 10 line 31-page 12, line 6:

With reference next to **Figure 4**, a diagram of a menu system is depicted in accordance with a preferred embodiment of the present invention. In this example, menu **400** is an IDE tools menu presented in a GUI. As illustrated, menu **400** is a drop down menu, which may be displayed by spell checking system **318** in **Figure 3** in providing spell checking features to a user. Spell check function **402** is located within this menu. When spell check function **402** is selected, a submenu **404** is presented containing "check from open source file" function ~~406~~, "check external resource file" function 406 ~~408~~, "add user defined word to dictionary" function 408 ~~410~~, and options function **412**.

"Check from open source file" function 404 ~~406~~ allows the user to perform the spell checking function in the source file currently being worked on in the IDE. Selection of "check external resource file" function 406 ~~408~~ allows the user to select another file other than the file being worked on in the IDE for spell checking. This function allows a user to specify the location of the file. In this example, popup dialog **414** is displayed in response to selection of "check external resource file" function 406 ~~408~~. The user may enter a file path to the external resource file in field **416**. Other location information, such

as a universal resource identifier (URI) or a universal resource locator (URL), may be used to identify the location of the external resource file. Alternatively, instead of entering the file path name or location, the user may identify this location by selecting browse button **418**, which results in a tree being presented to the user. This tree may be traversed by the user to identify a location of the external resource file. When the location is correct, the user may select okay button **420**. The user may cancel this function by selecting cancel button **422**.

Page 4 line 29-page 5, line 4:

Figures ~~7~~ 7A-7C ~~is~~ show a diagram illustrating an example of a Java language resource file in accordance with a preferred embodiment of the present invention; and

Figures 8A-8C ~~and 8B is~~ show an example of C language resource file in accordance with a preferred embodiment of the present invention.

Page 14, line 25-page 15 line 11:

With reference now to **Figures 7A and 7B** ~~**Figure 7**~~, a diagram illustrating an example of a Java language resource file is depicted in accordance with a preferred embodiment of the present invention. Resource file **700** is an example of an external resource file that may be processed using the mechanism of the present invention.

Turning next to **Figures 8A** ~~**and 8B**~~, and **8c** an example of C language resource file is depicted in accordance with a preferred embodiment of the present invention. Resource file **800** is another example of resource file that may be processed by using the mechanism of the present invention. The locations of these resource files are identified in these examples by a user input containing a path or other universal resource locator. Of course the mechanism of the present invention also may use pointers to a resource file in the source code to locate the resource file for spell checking.